

Visualization Framework 的用户

本节将对我们所讨论的 Visualization Framework 进行用户层次分析，并讨论现有可视化工具和组件的用户层次问题。

为什么讨论用户

这里首先将明确本节对 Visualization Framework 用户进行讨论的意义。由于我们的项目没有确定的需求方，因此我们需要将框架所涉及、以及今后论述时所涉及的用户群体进行定义、描述和分析，并通过这种定义和分析以构建可视化框架需求。

用户定义与分析

能力类型

根据用户群的技能、期望界面和期望操作等方式，我们将用户能力分成以下四类。需要说明的是，对于一个真正的用户（人）来说，他/她可能具备以下四类能力中的一种或多种能力。

用户能力类型	技能	示例用户	示例行为	期望逻辑抽象	期望实际接口/界面 ¹
Core Programmer	<ul style="list-style-type: none">* 能够使用编程语言编写程序²* 透彻理解可视化流水线模型* 有能力修改、创建可视化流程模型	@ 可视化底层框架研究者 @ 开发维护可视化框架的人（a.k.a VAG 小组成员）	修改可视化流水线 创建诸如 Semantic Interaction 的可视化流水线扩展	（无）需要自行构建	（编程语言）
Programmer	<ul style="list-style-type: none">* 能够使用编程语言编写程序* 了解可视化流水线模型* 能够编写针对流水线中某个阶段的基本功能模块	@ 论文投稿者 @ 创建新视图、新变换算法、新交互的研究者	编写读取某数据格式的 parser 编写一个新的投影算法模块	整个可视化流水线的抽象模型	编程 API
Scripter	<ul style="list-style-type: none">* 能够理解脚本语言* 理解框架提供的布局、交互接口* 可以创建	@ 原 Processing、D3 用户 @ Designer 中希望创建复杂布局和图元的用户	创建已有布局算法模块的组合（例如，在 nodelink 的每个节点上放一个 treemap）	<ul style="list-style-type: none">* 流水线抽象模型* 已有视觉映射、布局、交互模块的抽象* 构建新模块、	* 脚本系统

¹这些接口/界面由上一层次提供。
²这里所指的“编程语言”即是用于构建可视化框架的语言。下方表格中的该条目具有与此相同的意义。

				组合型模块的方法	
Designer	* 没有程序经验 * 有数据基础 * 懂得如何设计并构建可视化应用	@ 数据新闻记者 @ 商业报表制作人员 @ 平面设计人员	将收集的数据制作为报表 制作最终的可视化应用 制作信息图	画布 基础图形 数据浏览 数据与图形的绑定操作	交互式界面（所有操作使用鼠标键盘等即可完成）

表 1：用户能力类型

一些附加说明：

- 四种用户能力类型并不相互排斥，只是代表了在可视化方面拥有不同能力层次的用户。
- 一个用户可具有多种用户能力。例如，一个使用交互式界面的用户（Designer 能力）也可以使用脚本完成一些复杂的布局（Scripter 能力）。
- 拥有该能力所需的可视化专业程度：由 Core Programmer 至 Designer 依次递减；
 学习成本：由 Core Programmer 至 Designer 依次递增；
 对可视化程序的控制自由度：由 Core Programmer 至 Designer 依次递减。

这里我们将用户能力分成两种层次：应用级（**Application Level**）和开发级（**Development Level**），其中 Core Programmer 和 Programmer 分为开发级，Scripter 和 Designer 分为应用级。两种层次的主要区别是，应用级能力用户使用现有功能构建可视化应用，而开发级能力用户主要关注如何构建应用级能力用户使用的平台。